Biosystems and Information Technology (2013) Vol.2(2) 32-36
DOI: http://dx.doi.org/10.11592/bit.131001

original scientific article

Hybrid discrete algorithm for the modelling of gene regulatory networks

Dušan Vučko[1,2*], Miha Mraz[1], Nikolaj Zimic[1], Miha Moškon[1]

[1]*Faculty of Computer and Information Science, University of Ljubljana, Slovenia*
[2]*ComTrade d.o.o., Letališka cesta 29b, 1000 Ljubljana, Slovenia*

**Corresponding author*
dusan.vucko@comtrade.com

Received: 31 October 2013; accepted: 26 November 2013; published online: 28 November 2013.
This paper has no supplementary material.

Abstract: *Mathematical modelling and simulation can aid the analysis and design of gene regulatory networks (GRNs). GRN modelling approaches can be divided into two major categories, deterministic and stochastic. In this paper we present a new algorithm for GRN modelling called hybrid discrete algorithm (HDA). It introduces stochastic effects into an underlying deterministic approach and is based on implicit rules that make modular, bottom-up modelling possible, without having to derive specific network equations. The algorithm explicitly models competitive binding of activators and repressors to the same binding site. Furthermore, it takes into account a limited number of binding site repeats. We demonstrate and validate the algorithm on the repressilator model.*

Keywords: Gene regulatory networks, gene expression modelling, competitive binding, fractional occupancy, repressilator.

1. Introduction

Gene regulatory networks (GRNs) play a central role in synthetic biology as they enable the modification of existing and realization of novel cellular logic (Voigt, 2006). While many approaches for GRN modelling exist, two major ones are deterministic and stochastic (Kaern, et al., 2003). Deterministic models can be based on ordinary differential equations (ODEs), which make the use of analytical techniques possible. Stochastic models are, on the other hand, typically established with a Chemical Master Equation (CME), which can be solved numerically with different approaches, such as a stochastic simulation algorithm (SSA) (Gillespie, 1976). Deterministic models can be used to describe a time average of cell population dynamics, while computationally more demanding stochastic models are needed to capture the behaviour of a single cell (Kaern, et al., 2003). The dynamics obtained with the deterministic and stochastic models can converge if mRNA and protein concentrations are high, cell volumes are large and promoter kinetics are fast (Kaern, et al., 2005).

Usually, these models are formulated on a per-GRN basis using a top-down approach. Furthermore, phenomena such as transcription factor binding and competitive binding are often not explicitly modelled. To address some of these issues, we present the hybrid discrete algorithm (HDA) for GRN modelling. The algorithm assumes high mRNA and protein concentrations and that transcription factor binding is much faster than transcription and translation, which is often the case in observed systems.

2. Hybrid discrete algorithm

2.1. Algorithm overview

HDA is a hybrid algorithm because it introduces the stochastic effects to an otherwise fundamentally deterministic gene expression modelling approach. The algorithm is discrete in a sense that it represents the species concentrations, which are evaluated in discrete time steps, as integers. In contrast, deterministic ODE models typically operate with real numbers in continuous time and space (Shmulevich and Aitchison, 2009).

Modelling using HDA consists of two major steps. First, each GRN is defined as a collection of its fundamental building blocks, or *entities*. These are genes, promoters, transcription factor binding sites, mRNA molecules and protein molecules that can also act as transcription factors. Each entity has a set of parameters that define its behaviour, e.g. gene transcription rate or protein degradation rate. Relationships such as gene repression or activation are also specified. A simulation of the GRN dynamics is carried out as a finite sequence of discrete time steps. Internal rules are used to determine the number of mRNA and protein entities at each time step. Specifically, at each time step:

- the effects of potential input signals are considered (e.g. the presence of molecular signals may facilitate transcription factor binding);
- transcription factors bind to their target binding sites;
- transcription, translation and species degradation occur.

This approach allows the modular GRN modelling and can be implemented using object-oriented programming, where each entity exists as an individual object. Formulation of specific GRN system equations or chemical reactions can thus be avoided.

2.2. Data model

The algorithm was implemented in C#. The following classes are defined: *mRNA*, *Protein*, *Product*, *BindingSite*, *Promoter* and *Gene*.

Classes *mRNA* and *Protein* define an individual mRNA and protein entity, respectively. They have the following attributes:

- *int* t_B – entity birth time, i.e. time step when the entity was generated;
- *bool Alive* – when this value is *true*, the entity is active in the system; the value is set to *false* once the entity has been marked for degradation.

Class *Product* associates mRNA species with certain protein species and has the following attributes:

- *List<mRNA> mRNAs* – a list of mRNA entities; newly transcribed mRNA entities are added to this list and removed from it once marked for degradation;
- *List<Protein> Proteins* – a list of protein entities; newly translated protein entities are added to this list and removed from it once marked for degradation;
- *int T* – translation rate, i.e. the number of protein entities to produce per mRNA entity;
- *int* τ_D – transcription-translation delay, i.e. the number of time steps that must elapse between transcription and translation;
- *double* q_M – mRNA degradation rate, i.e. the percentage of active mRNA entities to degrade at each time step;
- *double* q_P – protein degradation rate, i.e. the percentage of active protein entities to degrade at each time step.

Class *BindingSite* defines an individual transcription factor binding site and has the following attributes:

- *int C* – binding site capacity, i.e. the number of binding site repeats;
- *int* B_A – the number of activator entities bound to this binding site;
- *int* B_R – the number of repressor entities bound to this binding site.

Class *Promoter* defines an individual promoter. It has the following attributes:

- *PromoterType Type* – a promoter type that can be either minimal or constitutive;
- *List<BindingSite> BindingSites* – a list of binding sites associated with the promoter;
- *int* K_A, K_R, b_0, b_1, *k, z, a, r* – constants that regulate transcription rates of genes associated with the promoter (see 2.4. for complete description).

Class *Gene* defines a gene and has the following attributes:

- *Promoter* P_G – gene promoter;
- *double n, m* – non-linearity coefficients used for transcription modelling.

2.3. Modelling the binding of transcription factors

HDA can explicitly model competitive binding of an activator and a repressor to the same binding site that affects the corresponding promoter activity. Let *B* be a binding site with *C* repeats, i.e. with capacity *C*. A single activator or repressor entity can bind to each binding site repeat. The sum of all repressor and activator entities bound to a binding site is never greater than the binding site's capacity:

$$B_A + B_R \leq C. \quad (1)$$

Suppose that in a time step *t*, the number of activators and repressors that bind competitively to the same binding site is *A* and *R* respectively. If $A + R \geq C$, they are distributed among the available binding site repeats according to the equations:

$$B_A = C \cdot \frac{w_A \cdot A}{w_A \cdot A + w_R \cdot R}, \quad (2)$$

$$B_R = C \cdot \frac{w_R \cdot R}{w_A \cdot A + w_R \cdot R}, \quad (3)$$

where w_A and w_R are the weights specifying the activator and repressor binding affinity, respectively. Note that B_A and B_R are integers, hence rounding is used. If we assume equal binding affinity, then $w_A = w_R = 1$ and a uniform distribution of competing transcription factors is obtained. This way, the amount of bound transcription factors is proportional to their available concentrations (i.e. the number of all existing entities). If $A + R < C$, all available activator and repressor entities can bind to the available binding site repeats, i.e. $B_A = A$ and $B_R = R$. In case of non-competitive binding, $A = 0$ if only repressor binds to *B*; similarly, $R = 0$ if only activator binds.

2.4. Gene expression modelling

To each gene in a GRN, a promoter and a list of mRNA entities are assigned. The list, which contains all mRNA entities that exist at a specific point, can be shared among multiple genes. Each mRNA species is associated with a specific protein species represented as a list of protein entities. When an mRNA entity is translated, a new protein entity is added to the list of protein entities. A transcription-translation delay can be specified as a number of time steps that must elapse after an mRNA entity has been generated and before the corresponding protein is generated.

The gene transcription rate is regulated by the binding of transcription factors to the binding sites associated with their promoters. HDA presumes two different types of promoters, minimal and constitutive. Binding of an activator is required to achieve a significant increase of transcription rate of genes regulated by a minimal promoter, as RNA polymerase has low binding affinity for it. In contrast, genes under a constitutive promoter are transcribed even in the absence of transcription factors. Binding of a repressor, however, decreases the transcription rate, ideally to zero, effectively turning the genes off. However, in realistic experimental settings, a certain amount of leaky transcription is present despite the bound repressor.

Transcription is modelled as follows. At each simulation time step *t*, *activation intensity* A_N and *repression intensity* R_N are calculated for each gene:

$$A_N = \sum_i B_{A_i}, \quad (4)$$

$$R_N = \sum_i B_{R_i}, \quad (5)$$

where B_{A_i} and B_{R_i} are the total number of bound activator, respectively repressor, entities on the gene promoter's binding site *i*. Hence, activation, respectively repression, intensity is the sum of all activator, respectively repressor, entities bound to gene promoter's binding sites.

Next, the number of mRNA entities (N'_{mRNA}) to generate at time step *t* is determined for each gene in the GRN. Note that rounding is used as N'_{mRNA} is a non-negative integer. Two distinct situations are possible based on the promoter type. In case of a minimal promoter:

$$N'_{mRNA} = b_0 + b_1 - b_1 \cdot \frac{r \cdot R_N^n}{K_R^n + r \cdot R_N^n} + k \cdot \frac{a \cdot A_N^m}{K_A^m + a \cdot A_N^m + r \cdot R_N^n} \quad (6)$$

where:

- b_0, b_1 and k are transcription rate constants of genes regulated by the minimal promoter;
- r is an association constant between a repressor and a binding site;
- a is an association constant between an activator and a binding site;
- n and m are non-linearity coefficients;
- K_R and K_A are constants that specify repression and activation threshold, respectively – the quantity of activator and repressor entities required to achieve a certain transcription rate.

Leaky transcription rate (b_0) represents the minimal transcription rate that is present even when the promoter is strongly repressed. Ideally, $b_0 = 0$. When no transcription factors are bound (i.e. $A_N = R_N = 0$), transcription rate equals $b_0 + b_1$. Here, b_1 is transcription rate that can be eliminated by a repressor binding. Although low transcription is expected, binding of a repressor can further decrease the transcription rate. Maximal transcription rate attainable is $b_0 + b_1 + k$ and is reached only when a sufficient amount of activator entities and no repressor entities are bound to a promoter's binding sites. Normally, we assume $b_0 \ll k$ and $b_1 \ll k$. In the absence of transcription factors, each gene under a minimal promoter will be transcribed at a relatively low rate that equals $b_0 + b_1$ (ideally 0, i.e. no transcription takes place at all).

The described transcription modelling approach is inherently deterministic and stems from a basic fractional occupancy model of gene expression (Sauro, 2012). To derive equation (6), let us assume a minimal promoter with a single binding site repeat to which either an activator or a repressor entity can bind exclusively. The promoter is in an active state, i.e. state leading to transcription of genes under its control, only when an activator is bound. Three states are possible:

- U – neither an activator nor a repressor is bound to the binding site (binding site is unoccupied), hence the promoter is inactive;
- U_R – a repressor R is bound to the binding site, hence the promoter is inactive;
- U_A – an activator A is bound to the binding site, hence the promoter is active and transcription occurs.

The probability of an active promoter is expressed as its fractional occupancy:

$$f = \frac{U_A}{U + U_A + U_R}. \quad (7)$$

Transitioning between the three promoter states can be described with the reactions:

$$U + A \underset{\beta_1}{\overset{\alpha_1}{\rightleftarrows}} U_A, \quad (8)$$

$$U + R \underset{\beta_2}{\overset{\alpha_2}{\rightleftarrows}} U_R, \quad (9)$$

Assuming that binding and unbinding of transcription factors occurs at a much higher rate than transcription, equilibrium is reached. According to the law of mass action, we write:

$$\alpha_1 \cdot U \cdot [A] = \beta_1 \cdot U_A, \quad (10)$$

$$\alpha_2 \cdot U \cdot [R] = \beta_2 \cdot U_R. \quad (11)$$

It follows that:

$$U_A = \frac{\alpha_1}{\beta_1} \cdot U \cdot [A] = a \cdot U \cdot [A], \quad (12)$$

$$U_R = \frac{\alpha_2}{\beta_2} \cdot U \cdot [R] = r \cdot U \cdot [R]. \quad (13)$$

Fractional occupancy can now be expressed as:

$$f = \frac{a \cdot U \cdot [A]}{U + a \cdot U \cdot [A] + r \cdot U \cdot [R]} = \frac{a \cdot [A]}{1 + a \cdot [A] + r \cdot [R]}, \quad (14)$$

which we generalize to:

$$f = \frac{a \cdot [A]^m}{K_A^m + a \cdot [A]^m + r \cdot [R]^n}. \quad (15)$$

If we multiply the obtained expression with a constant k, which represents maximal attainable transcription rate, we can – in the context of HDA – interpret the result as a number of mRNA entities to produce in a time step:

$$N'_{mRNA} = k \cdot \frac{a \cdot A_N^m}{K_A^m + a \cdot A_N^m + r \cdot R_N^n}, \quad (16)$$

which is equal to the last term in (6). Even if leaky transcription rate b_0 is introduced as an additional term, the problem with formulation (16) is that in general, it doesn't distinguish between a transcription rate when no transcription factors are bound (i.e. $b_0 + b_1$) and a transcription rate when no activator is bound and the promoter is fully repressed (i.e. b_0). Both situations can be represented simply as an inactive promoter. For this reason, we include the additional terms:

$$b_1 - b_1 \cdot \frac{r \cdot R_N^n}{K_R^n + r \cdot R_N^n} \quad (17)$$

that become relevant especially if the difference between b_0 and b_1 is relatively large. Note that this can be rewritten as the equation

$$b_1 \cdot \left(1 - \frac{r \cdot R_N^n}{K_R^n + r \cdot R_N^n}\right) = b_1 \cdot \frac{K_R^n}{K_R^n + r \cdot R_N^n}, \quad (18)$$

which equals the Hill function of a repressor if $r = 1$ and $R_N = [R]$ denotes repressor concentration.

We have described the HDA implementation of transcription model for genes regulated by a minimal promoter (equation (6)). However, HDA can also model constitutive promoters. The number of mRNA entities to generate for each gene regulated by a constitutive promoter is calculated as:

$$N'_{mRNA} = b_0 + k - k \cdot \frac{r \cdot R_N^n}{K_R^n + r \cdot R_N^n} + z \cdot \frac{a \cdot A_N^m}{K_A^m + a \cdot A_N^m + r \cdot R_N^n}, \quad (19)$$

where b_0 is a leaky transcription rate and k is normal, constitutive transcription rate that takes place when no repressor is bound. Bound repressor entities can significantly lower the transcription rate and result in $k = 0$. If activator binding sites are also associated with the constitutive promoter, binding of an activator can increase the transcription rate for a maximum of z. Hence, maximal transcription rate is $b_0 + k + z$. Normally, we assume $b_0 \ll k$.

Once N'_{mRNA} has been determined, it is - regardless of the promoter type - either increased or decreased by δ%:

$$N_{mRNA} = N'_{mRNA} + \delta \cdot N'_{mRNA}, \quad (20)$$

where N_{mRNA} is the final amount of mRNA entities to produce for a given gene at step t , δ is uniformly distributed random variable from an interval $[-\lambda, \lambda]$ and λ is *transcription stochasticity* percentage.

Translation is modelled probabilistically, i.e. at each simulation step, for each existent mRNA entity, T protein entities are produced with *translation probability* P_T (a uniform distribution is used) if a delay between transcription and translation has already elapsed.

Protein and mRNA entities have a *degradation rate* parameter. At each time step, q_M% of existent mRNA entities and q_P% of existent protein entities are degraded, i.e. removed from the system, where q_M is mRNA degradation rate and q_P is protein degradation rate.

3. Sample model (repressilator)

The repressilator (Fig. 1) can be realized with a GRN consisting of three genes that mutually repress one another: each gene encodes a repressor for another gene (Elowitz and Leibler, 2000). Conditions exist where concentrations of the three repressors oscillate. We model the repressilator using HDA with the following experimental parameter values: each binding site has a capacity $C = 20$. Each gene under a constitutive promoter has a transcription rate constant $k = 10$; no leaky transcription is assumed ($b_0 = 0$). Translation rates of mRNA species are $T = 1$. Non-linearity coefficient is $n = 2$, with constants $K_R = 4$ and $r = 1$. Degradation rates are $q_M = 0.45$ for mRNA species and $q_P = 0.1$ for protein species. Transcription stochasticity is set to $\lambda = 0.75$ and translation probability to $P_T = 0.9$. Parameter values are chosen in a way to comply with theoretical requirements for oscillatory behaviour, namely strong promoters, high non-linearity coefficient and low leakiness. Activation-related parameters are irrelevant since no activators are present in the system. No transcription-translation delay is assumed. Simulation results of the model are shown in Fig. 2 and capture the main dynamics (i.e. oscillatory behaviour) in accordance with deterministic models in the relevant literature.

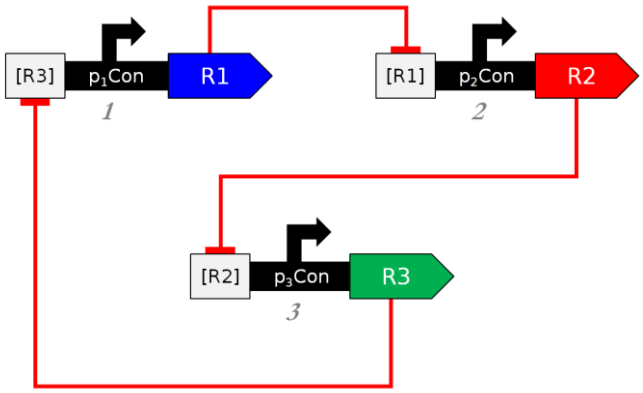

Fig. 1. **Repressilator consists of three genes (R1, R2, and R3) under constitutive promoters (p_1Con, p_2Con, p_3Con) with a single repressor binding site ([R1], [R2] and [R3]). Concentrations of repressors encoded by R1, R2 and R3 can oscillate under certain conditions.**

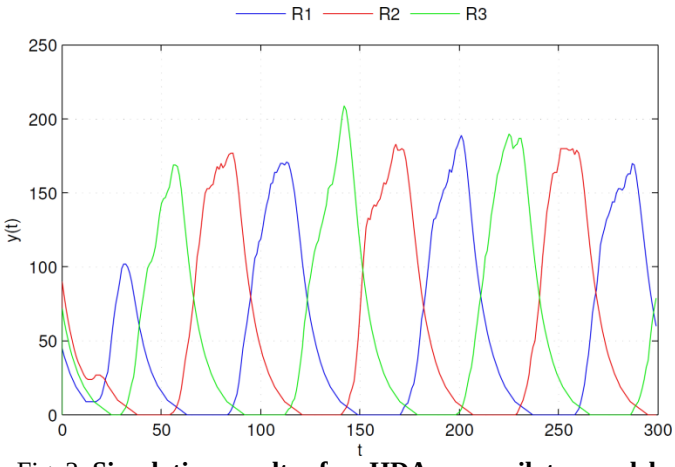

Fig. 2. **Simulation results of an HDA repressilator model demonstrating oscillatory behaviour. Number of repressor protein entities is shown as a function of time (i.e. discrete time steps). Initial amounts of repressors are $R1 = 50$, $R2 = 100$ and $R3 = 80$. No absolute parameter units are used - obtained characteristics, such as species concentrations and a period of oscillations, must thus be interpreted in relative terms.**

4. Conclusion

The introduced hybrid discrete algorithm is suitable for modelling of GRNs where explicit formalization of transcription factor binding is desired, such as competitive binding of an activator and a repressor to the same binding site, which is crucial for implementing desired cellular logic in some networks. The algorithm enables modular, bottom-up modelling of GRNs and is designed with object-oriented programming implementation in mind.

While the algorithm uses stochastic elements, it is deterministic at its core, unlike inherently stochastic gene expression in realistic cellular environments. For this reason, the algorithm is only suitable for describing major GRN characteristics under deterministic modelling assumptions. It should also be noted that the output of the algorithm is highly dependent on the parameter values, evaluation of which may often be difficult due to e.g. lack of experimental data.

Acknowledgements

The research was partially supported by the scientific-research programme Pervasive Computing (P2-0359) financed by the Slovenian Research Agency in the years from 2009 to 2013. We acknowledge the members and mentors of the 2012 Slovenian iGEM Team for the motivation in the development and application of introduced modelling methodology. Project details are available on http://2012.igem.org/Team:Slovenia.

References

Elowitz, M. B. and Leibler, S., (2000). A synthetic oscillatory network of transcriptional regulators. *Nature,* Volume 403, pp. 335-338. http://dx.doi.org/10.1038/35002125

Gillespie, D. T., (1976). A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *Journal of Computational Physics,* Volume 22, pp. 403-434. http://dx.doi.org/10.1016/0021-9991(76)90041-3

Kaern, M., Blake, W. J. and Collins, J., (2003). The Engineering of Gene Regulatory Networks. *Annual Review of Biomedical Engineering,* Volume 5, pp. 179-206. http://dx.doi.org/10.1146/annurev.bioeng.5.040202.121553

Kaern, M., Elston, T. C., Blake, W. J. and Collins, J. J., (2005). Stochasticity in Gene Expression: From Theories to Phenotypes. *Nature Reviews Genetics,* Volume 6, pp. 451-464. http://dx.doi.org/10.1038/nrg1615

Sauro, H. M., (2012). *Enzyme Kinetics for Systems Biology.* s.l.:Future Skill Software (Ambrosius Publishing).

Shmulevich, I. and Aitchison, J. D., (2009). Deterministic and stochastic models of genetic regulatory networks. *Methods Enzymol.,* Volume 467, pp. 335-356. http://dx.doi.org/10.1016/S0076-6879(09)67013-0

Voigt, C. A., (2006). Genetic parts to program bacteria. *Current Opinion in Biotechnology,* Volume 17(5), pp. 548-557. http://dx.doi.org/10.1016/j.copbio.2006.09.001